

البدء مع

ADO.NET





أن ما دعاني لبدء هذا الشرح هو التطور المستمر للبرمجة وصعوبة أيجاد شرح وافي لعملية الاتصال ومحاكاة قاعدة

بيانات من حفظ واسترجاع وتحديث وبحث .. ألخ . فكل ما نجده هو الأمثله والمشاريع الشخصية الجاهزة وهي مفيدة

ولكن لا يكون أثرها كالدرس المشروح فأغلبيتنا يفتقد لأقل معلومه وأسرعها وأن تكون سهلة الفهم لكي يستوعبـــــها

المبتدء قبل المحترف كحالي عندما بدءت البرمجه والاحتراف كلمة لتمييز قدرات الشخص على الابتكار والتفكيـــــــر

والأبداع وهي في النهاية تفـــاوت في المستويات ولكنها ليست بدرجة الكمال لأن الكمال لله وحده

سبحانه وتعالـــــــــــــى

فالبرمجة بأعتقادي الشخصي بحر شاسع لا حدوده له لتنوعها .. فربما المبتدء يجد فيها ما يتوقف عنده المــــحترف

وهذا هو لب الموضوع

التعاون بأقل معلومة هو الاحتراف بحد ذاته



متطلبات المشروع

[1]برنامج Visual Studio .NET

[2]معرفة مسبقة بلغة الفيجول بيسك دوت نت والتعامل مع نماذجه وأداوته

[3]قاعدة بيانات (وهنا سنبدأ مع الأكسس ثم سنحولها إلى قاعدة سيرفر وسنرى الاختلاف(

[4]عليك بكتابة الشيفرات الموجودة في الدرس لكي تخطىء وتعرف أين أخطأت

ملاحظة : اتمنى من الاخوة الاعضاء عند أدراجكم لمشاركاتكم أن تقتصر على الأسئلة والأستفسارات بعكس المدح والشكر .. فكل مانريده هو الأفادة والأستفادة وعدم أكثار الصفحات بدون فائدة .. ومن يريد أبدء أعجابه أو أنتقاده للموضع أن يراسلني برسالة على الخاص.

نبذه عن ADO.NET

هي مجموعة من الفئات مشمولة في مجال الأسماء **System.Data** غرضها الوصول إلى مصادر البيانات **Data Sources** والتي تمثل بيانات محفوظة تحت أنظمة قواعد بيانات متعددة الأنواع مما يعني قدرتك على الوصول إلى أي قاعدة بيانات مهما كانت الشركة المنتجة لها

)أ. تركي العسيري (

الأختلافات الجوهرية بين ADO and ADO.NET

** ADO **

1-مصمة للعمل في بيئة متصلة بأستمرار مع قاعدة البيانات

2-يستخدم الكائن RecordSet للأحتفاظ بمجموعة بيانات واحدة

3-تحتوي على أنواع من المؤشرات Cursors المستخدمة لأغراض مختلفة ولكل مؤشر أمكانياته الخاصة

4-تخزن البيانات في هيئتها الثنائية مما يصعب أرسالها عبر جدران الحماية . كما أنها غير مفيدة للأنظمة التي لا تدعمADO

5-تستهلك قدراً من موارد النظام بسبب اتصالها الدائم بقاعدة البيانات أثناء المعالجة

** ADO.NET **

1-مصممة من الأساس للعمل في بيئة غير متصلة) ويمكنها العمل باتصال دائم مع قاعدة البيانات(

2-يستخدم الكائن DataSet للأحتفاظ بعدة مجموعات من البيانات

3-لا تستخدم المؤشرات لأنها تعمل في بيئة غير متصلة.

4-تخزن البيانات في هيئة XML العالمية . وهذه الهيئة مصممة لكي ترسل عبر جدران الحماية وعبر الشبكات دون مشاكل كما يمكن لأي تطبيق قراءة البيانات بهيئة XML بسهولة.

5-تعمل كنظام بيانات منفصل عن قاعدة البيانات فهي لا تتصل بقاعدة البيانات إلا عند الضرورة وبالتالي لن تستهلك مورد النظام الإ عند الضرورة

والأختلافان الأخيران رقم 5 - 4 هما جوهرة الاختلاف وأهمها





<u>معماريةADO.NET</u>

والمقصود هنا الخصائص المزوده والمساعدة في عملية الاتصال لقراءة وتكييف البيانات وهي

-[1]مجموعة البياناتDataSet

وهو الكائن المكافىء للكائن RecordSet ولكن مع الكثير من المزايا والتحسينات حيث يستطيع تخزين أكثر من جدول أو نتيجة أستعلام في نفس الوقت حيث يمثل كل واحد من هذه الجداول كائناً

منفصلاً عن الآخر

-[2]مجموعة البيانات DataAdapter

يمثل الجسر الذي يربط بين DataSet وقاعدة البيانات ويدعم أوامر - Select - Update Delete - Insertوبالتالي بإمكانه القيام بعمليات مختلفة على البيانتا كما أنه المسؤول عن تحميل كائن DataSetبالبيانات

-[3]مجموعة البيانات DataReader

يستخدم هذا الكائن لقراءة البيانات فقط ويمكنه قراءة كميات ضخمه منها تلك التي لا يمكن تخزينها في الذاكرة مؤقتاً

-[4]مجموعة البيانات DataRelation

يستخدم هذا الكائن لتمثيل العلاقات بين الجداول في قاعدة البياناتJOIN

-[5]مجموعة البيانات Connection

يعمل هذا الكائن بصورة مشابهة للكائن Connection في ADO وهو يمكننا من إنشاء اتصال مع قاعدة البيانات

-[6]مجموعة البياناتCommand

يسمح هذا الكائن لكائن DataAdapter بتطبيق الأوامر على قاعدة البيانات ويمكنه أن يتضمن اربعة من هذه الأوامر

لأضافة أستفسار أو سؤال حول الموضوع من هنا

الدرس التالي في عملية الاتصال بقاعدة البيانات وكيف يتم ذلك

****

لخدمات تصميم وبرمجة المواقع

programmer4ever@yahoo.com

00201063879624

****

الدرس الثاني : عملية الاتصال بالقاعدة



تتطلب عملية الوصول إلى البيانات من خلال ADO.NET على نوع مصدر البيانات الذي تود الاتصال معه بالتصريح عن مزود البيانات لو فرضنا أننا نستخدم عملية الاتصال بالكود

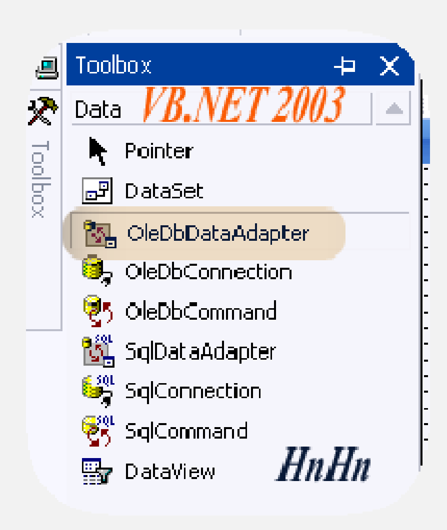
كود

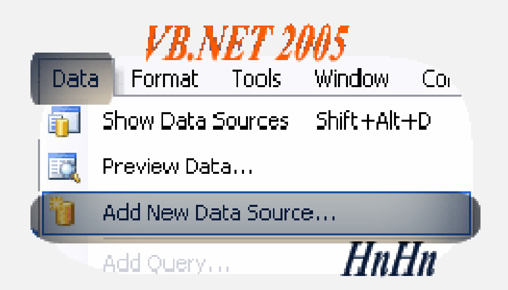
```
Dim cn As New OleDbConnection(connString)
cn.Open()
Dim rs As New OleDbDataAdapter("SELECT * FROM [ Table]", cn)
Dim rs As New DataSet()
rs.Fill(rs, "Table")
```

وسنأتي على شرح ما سبق بتفصيل حيث سأذكر اولا طريقة الاتصال بواسطة المعالج لأن هدفي الاساسي توضيح المسألة بأكبر قدر من المعلومات كما يلي

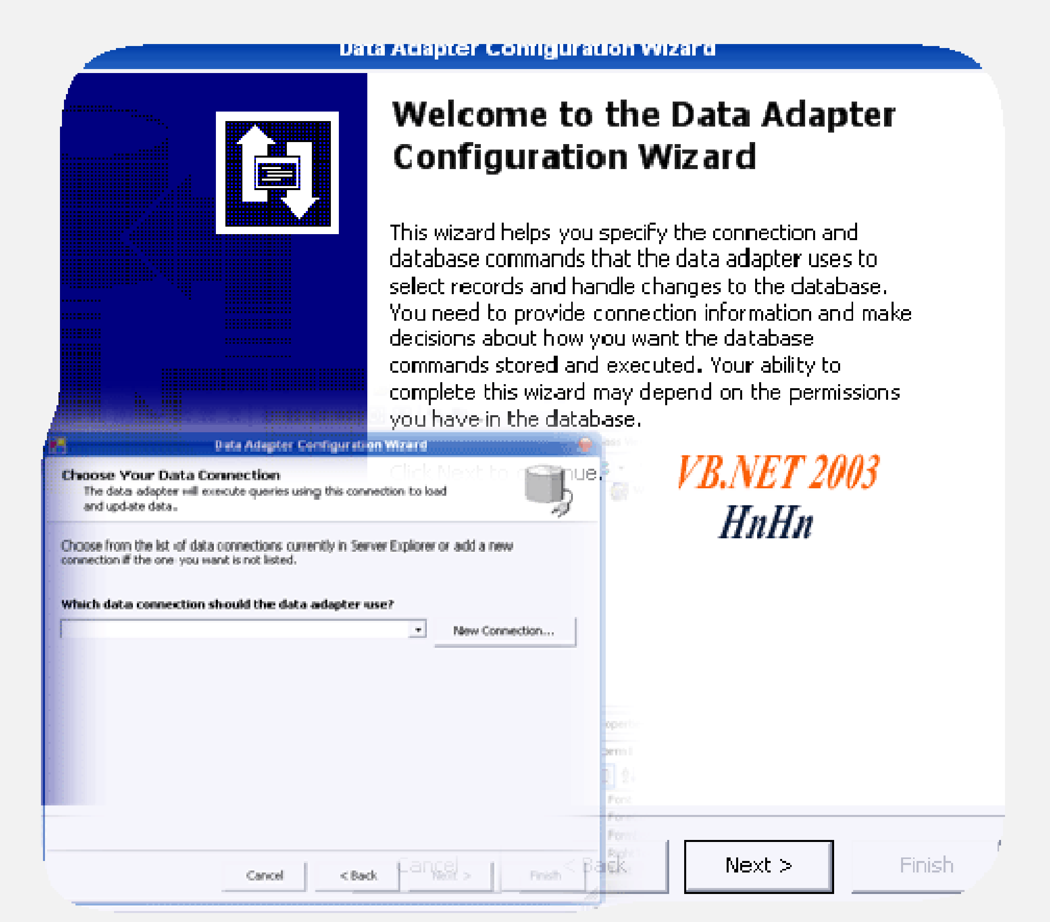
الخطوةالأولى : أنشاء كائن DataAdapter بواسطة المعالج في2003 - 2002

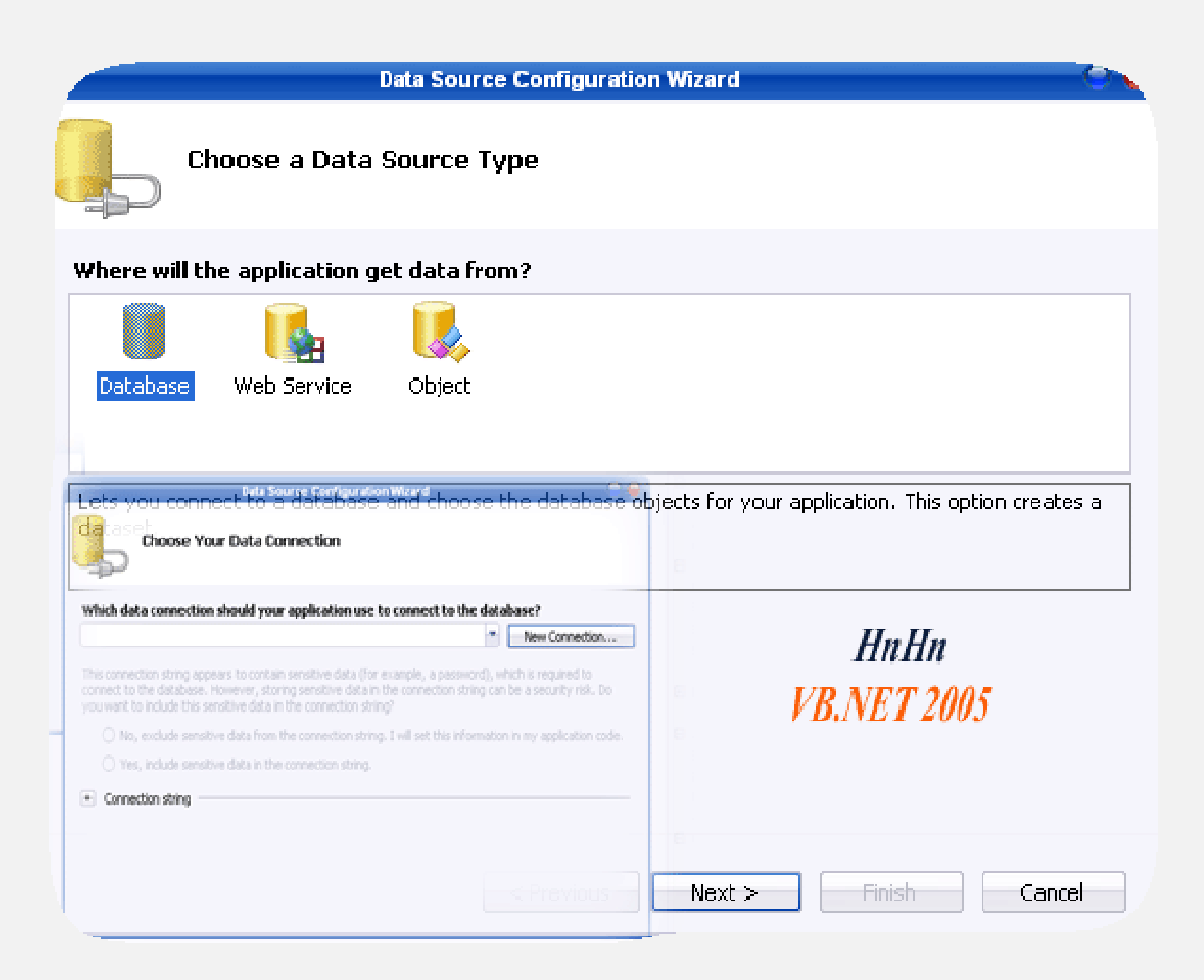
حيث ستقوم بتحديد الكائن OleDbDataAdapter ثم قم برسمه على الفورم اما في 2005 تم تحديد هذا الكائن من قائمةData ---> Add New Data Source





الخطوة الثانية : تحديد المزود ثم تحديد مصدر البيانات الذي تود الاتصال معه عن طريق انشاء اتصال جديد New Connection


Data Adapter Configuration Wizard
Welcome to the Data Adapter Configuration Wizard
This wizard helps you specify the connection and database commands that the data adapter uses to select records and handle changes to the database. You need to provide connection information and make decisions about how you want the database commands stored and executed. Your ability to complete this wizard may depend on the permissions you have in the database.
VB.NET 2003
HnHn
Choose Your Data Connection
The data adapter will execute queries using this connection to load and update data.
Choose from the list of data connections currently in Server Explorer or add a new connection if the one you want is not listed.
Which data connection should the data adapter use?
New Connection...
Cancel
< Back
Next >
Finish

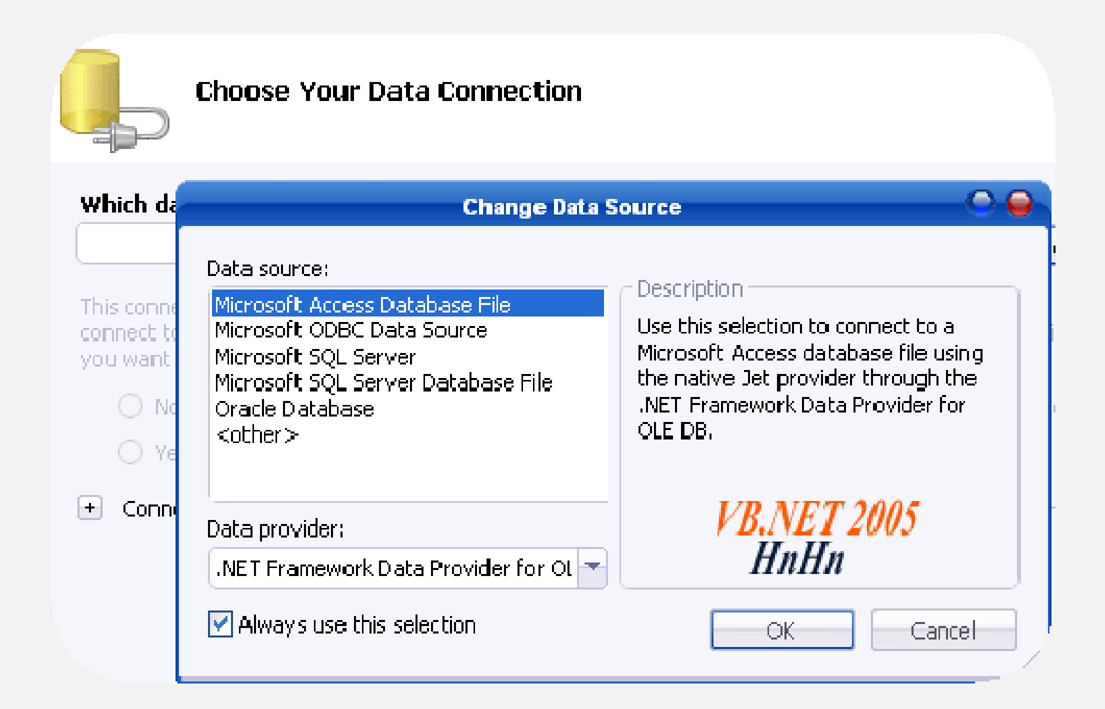


ثم نختار .. تبويب الموفر لتحديد المزود

حيث سنحدد المزود الخاص بالاكسس وبإمكانك اختيار المزود الخاص بـ SQL

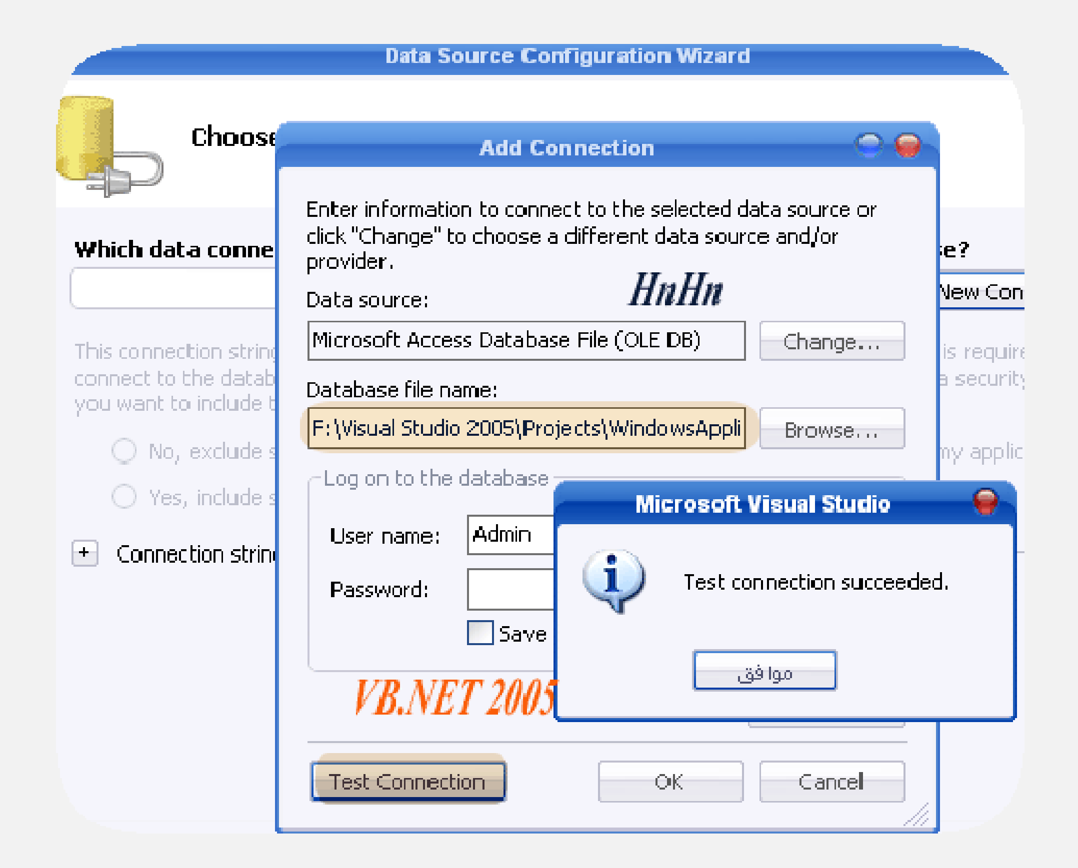

الموفر
الاتصال
خيارات متقدمة
الكل
HnHn
حدد البيانات التي تريد الاتصال بها:
موفر (موفري) OLE DB
MediaCatalogDB OLE DB Provider
MediaCatalogMergedDB OLE DB Provider
MediaCatalogWebDB OLE DB Provider
Microsoft Jet 3.51 OLE DB Provider
Microsoft Jet 4.0 OLE DB Provider
Microsoft OLE DB Provider For Data Mining Services
Microsoft OLE DB Provider for DTS Packages
Microsoft OLE DB Provider for Indexing Service
Microsoft OLE DB Provider for Internet Publishing
Microsoft OLE DB Provider for Microsoft Search
Microsoft OLE DB Provider for ODBC Drivers
Microsoft OLE DB Provider for OLAP Services
Microsoft OLE DB Provider for Olap Services 8.0
Microsoft OLE DB Provider for Oracle
Microsoft OLE DB Provider for Outlook Search
Microsoft OLE DB Provider for SQL Server
Microsoft OLE DB Simple Provider
VB.NET 2003

في2005



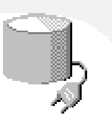
ثم قم بالضغط على زر اختبار الاتصال **Test Connection** لأختبار الاتصال


الكل
خيارات متقدمة
الاتصال
الموفر
حدد ما يلي للاتصال ببيانات Access:
١. حدد اسم قاعدة بيانات أو قم بإدخاله:
dowsApplication1\WindowsApplication1\DBHnHn.mdb
٢. أدخل معلومات تسجيل الدخول إلى قاعدة البيانات:
اسم المستخدم: Admin
كلمة المرور:
كلمة مرور فارغة
السماح بحفظ كلمة المرور
VB.NET 2003
Microsoft Data Link
HnHn
نجح اختبار الاتصال.
موافق
اختبار الاتصال


Data Source Configuration Wizard
Add Connection
Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.
HnHn
Data source:
Microsoft Access Database File (OLE DB)
Change...
Database file name:
F:\Visual Studio 2005\Projects\WindowsAppli
Browse...
Log on to the database
User name:
Admin
Password:
Save
VB.NET 2005
Microsoft Visual Studio
Test connection succeeded.
موافق
Test Connection
OK
Cancel

Choose Your Data Connection

The data adapter will execute queries using this connection to load and update data.



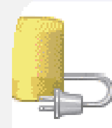
Choose from the list of data connections currently in Server Explorer or add a new connection if the one you want is not listed.

Which data connection should the data adapter use?

ACCESS.F:\Visual Studio 2005\Projects\WindowsApplication1\W | New Connection...

VB.NET 2003
HnHn

Cancel | < Back | Next > | Finish



Choose Your Data Connection

VB.NET 2005

HnHn

Which data connection should your application use to connect to the database?

ACCESS.F:\Visual Studio 2005\Projects\WindowsApplication1\WindowsApplicatior

New Connection...

Microsoft Visual Studio

The connection you selected uses a local data file that is not in the current project. Would you like to copy the file to your project and modify the connection?

If you copy the data file to your project, it will be copied to the project's output directory each time you run the application. Press F1 for information on controlling this behavior.

نعم   لا

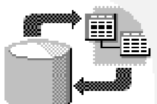
Connection string

Provider=Microsoft.Jet.OLEDB.4.0;Data Source=|DataDirectory|\DBHnHn.mdb

الخطوة الثالثة: كتابة الاستعلام SQL الذي يحدد البيانات كما سيأتي

Choose a Query Type VB.NET 2003

The data adapter uses SQL statements or stored procedures.



How should the data adapter access the database?

(•) **Use SQL statements**

Specify a Select statement to load d... and Delete statements to save da...

( ) **Create new stored procedures**

Specify a Select statement, and the ... select, insert, update, and delete rec...

( ) **Use existing stored procedures**

Choose an existing stored procedure ... delete).

Data Adapter Configuration Wizard

Generate the SQL statements

The Select statement will be used to create the Insert, Update, and Delete statements.

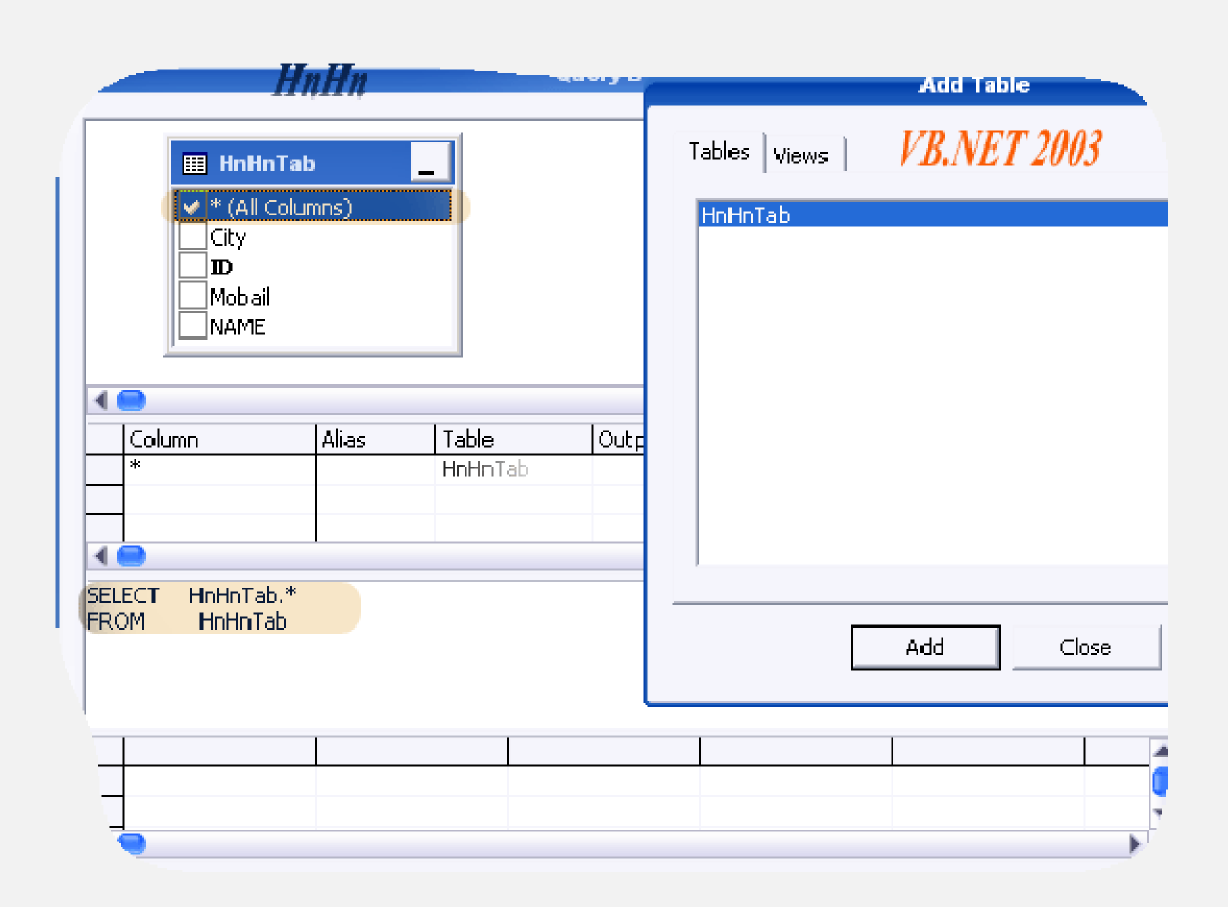
Type in your SQL Select statement or use the Query Builder to graphically design the query.

What data should the data adapter load into the dataset?

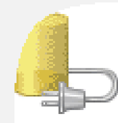
Advanced Options... | Query Builder...

HnHn

Cancel | < Back | Next > | Finish



أما في ٢٠٠٥ تتم العملية تلقائياً



Save the Connection String to the Application Configuration File

Storing connection strings in your application configuration file eases maintenance and deployment. To save the connection string in the application configuration file, enter a name in the box and then click Next.

Do you want to save the connection string to the application configuration file?

☑ Yes, save the connection as:

DBHnHnConnectionString

VB.NET 2005

HnHn

Data Source Configuration Wizard

Choose Your Database Objects

Which database objects do you want in your dataset?

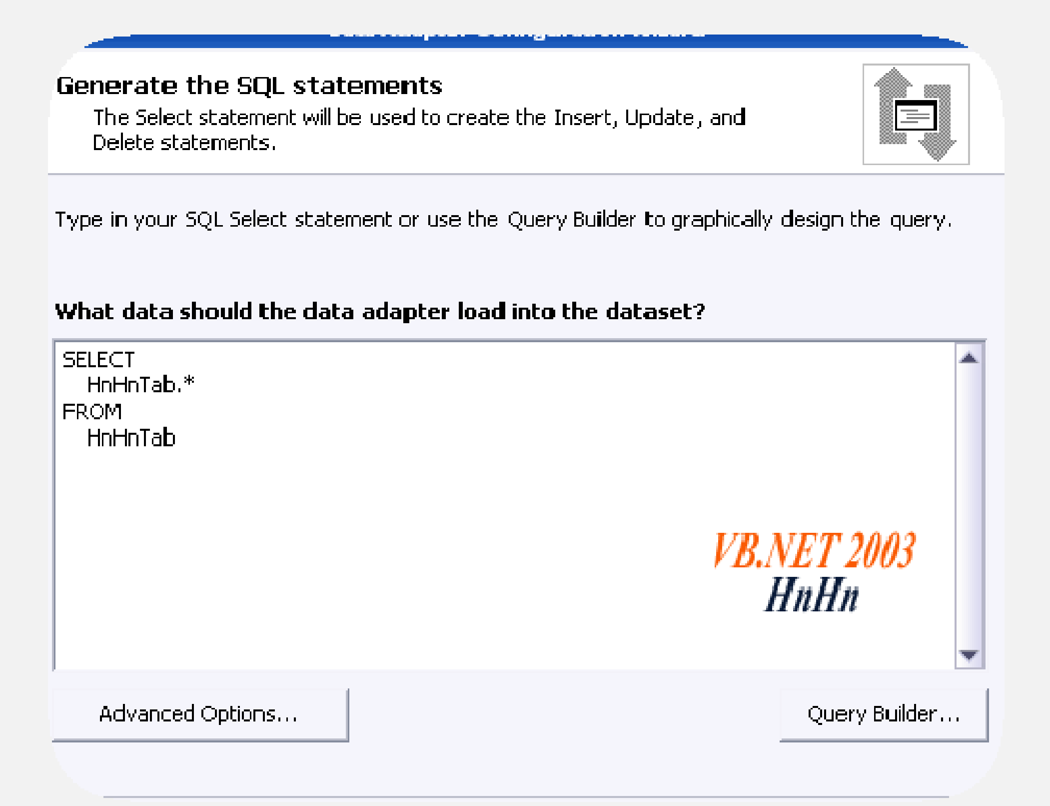
- ☑ Tables
  - ☑ HnHnTab
    - ☑ ID
    - ☑ NAME
    - ☑ City
    - ☑ Mobail
- ☐ Views

DataSet name:

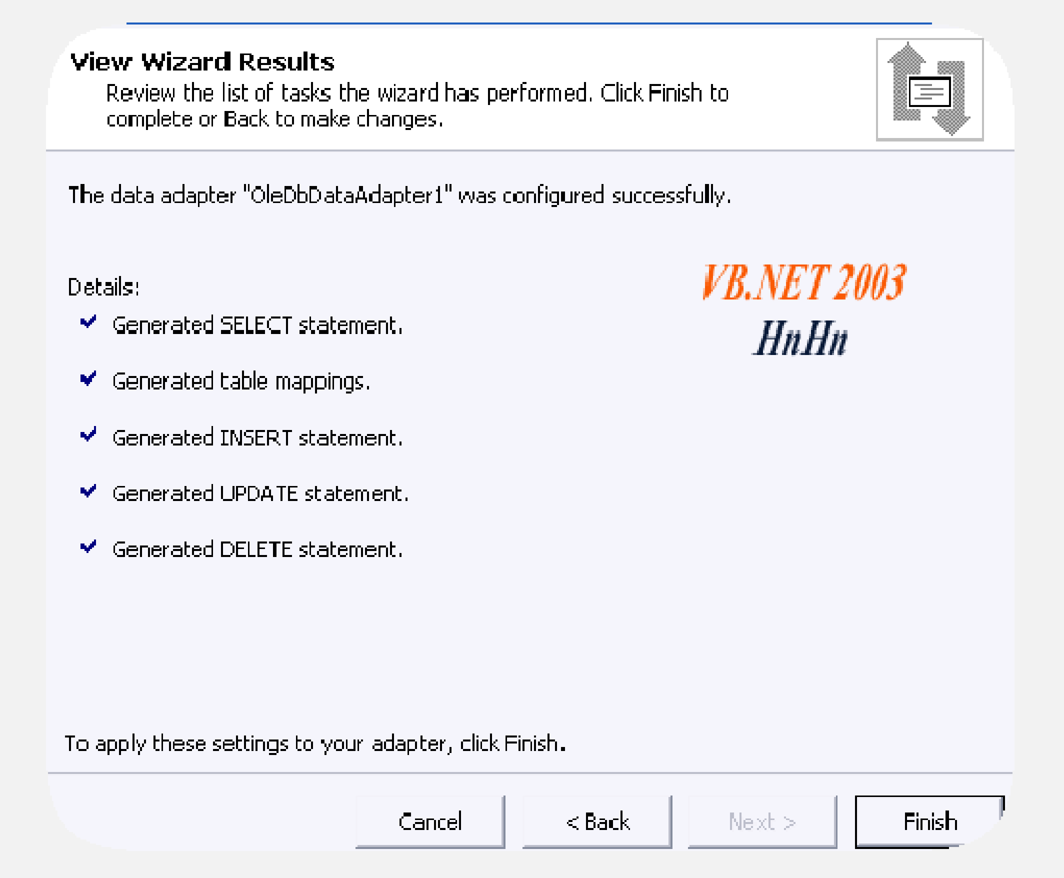
DBHnHnDataSet

< Previous | Next > | Finish

< Previous | Next > | Finish | Cancel

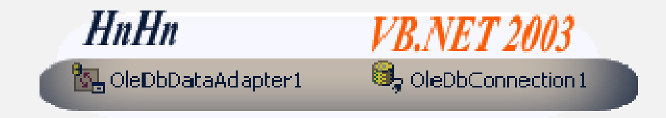


وأخيرا سيظهر إطار أخير يبين فيه العمليات التي تمت وتوليد أوامر الأضافة والتحديث والتعديل والحذف ونفس الشي مع2005



بعد الانتهاء من المعالج سيظهر لك الكائنان التاليين حيث أن المهالج قام بأنشاء كائن الاتصال بالأضافة إلى كائن

OleDbDataAdapter1





****

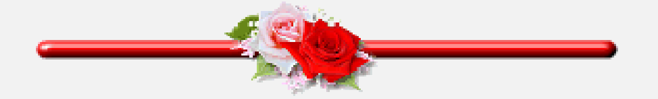
لخدمات تصميم وبرمجة المواقع

programmer4ever@yahoo.com

00201063879624

****

الأتصال بقاعدة البيانات عن طريق الأكواد Connection





وهذه الطريقة هي ما أفضلها دائما .. وستجد المتعة في ذلك من خلال تتبعك معنا لهذه الدروس وسوف يدور الكلام هنا حول الاتصال بقاعدة البيانات من نوع اكسس Access وذلك لأن الاغلبية يتعامل معها فلنبدأ . ...

قبل التعامل مع مصدر بيانات، عليك فتح اتصال معها وفي هذا القسم سنعرض الاساليب المتعددة للاتصال بمصادر البيانات.

تعريف كائن الاتصال

كود

```
Dim Con As New OleDbConnection()
```

حيث OleDbConnection : مخصصة للأتصال بقواعد البيانات من نوع OLE DB .NET Data Provider أما أذا كانت قواعد البيانات من نوع SQL Server .NET Data Provider فسيكون تعريف الاتصال من نوع SqlConnection و بهذا الشكل:

كود

```
Dim Con As New SqlConnection()
```

نص الاتصال بالقاعدة

سوف نقوم بتعريف متغير من نوع نص ووضع مسار الاتصال بقاعدة البيانات بداخله بهذه الطريقة

كود

```
Dim ConnString As String =
"Provider=Microsoft.Jet.OLEDB.4.0;Data Source
=c:\HnHnDB.mdb""
```

وهذ الكود يجب أسناده إلى مزود البيانات الذي يميل له كما سبق ذكره وهنا سيتم اسناده إلى OleDbConnection وستكون الشفرة بهذه الطريقة

كود

```
Dim con As OleDbConnection
con = New OleDbConnection(ConnString)
```

أو يمكنك كتابة التعريف بهذا الشكل ايضا

كود

```
Dim con As New OleDbConnection(ConnString)
```

ملاحظة :_بدلا من كتابة نص الاتصال كاملا في كل مرة تنوي انشاء كائن اتصال جديد قم بوضع نص الاتصال في متغير عام على مستوى المشروع

والقصد ان المتغير يختصر عليك كتابة شفرة الاتصال بهذه الطريقة في كل مرة تريد بها انشاء كائن للأتصال بالقاعدة لتحديث بياناتها.

كود

```
Dim con As New OleDbConnection
"Provider=Microsoft.Jet.OLEDB.4.0;Data Source
=c:\HnHnDB.mdb"
```

ولجعل المسار لمجلد القاعدة معروف تلقائيا كما في VB6 مثلاً والقصد هنا استخدام App.Path فقد تغييرت كلمة المسار التلقائي في الفيجول نت إلى Application.StartupPath وهناك أشكال عدة في استخراج المسارات في الفيجول نت سنذكرها لاحقاً أنشالله . وبهذه الكلمة ستكون شفرة الاتصال بهذه الطريقة

كود

```
Dim ConnString As String =
"Provider=Microsoft.Jet.OLEDB.4.0;Data Source =" &
Application.StartupPath & "\HnHnDB.mdb"
```

ملاحظة :_يجب أن تكون قاعدة البيانات **HnHnDB.mdb** في مجلد المشروع **Bin** يمكنك وضع القاعدة في اي مكان تريد ولكن الآن نحن نتكلم حول خاصية **Application.StartupPath** والتي تعرف مسار هذا المجلد من ضمن المشروع

أما بالنسبة لشفرة الاتصال بقاعدة بيانات من نوع سيرفر **SQL Server** وستكون بهذا الشكل وهي لا تختلف عن الاكسس سوى في عملية الاتصال
حيث أنك ستتعامل مع ال **Access** كملف لقاعدة بيانات اما **SQL Server** فأنك تتعامل مع محرك قواعد بيانات

كود

```
Dim SQLCon As New SqlConnection()
SQLCon.ConnectionString = "Data
Source=DEV4ARABS_SERVER;" _
& "User ID= HnHn ; Password= _ ";admin
القاعدة& "Initial Catalog= "
```

الآن سنأتي على فتح وإغلاق الاتصال بالقاعدة

بعد أن اسندنا نص الاتصال بالقاعدة للمتغير ConnString يمكننا الآن البدء بفتح الاتصال واغلاقه عن طريق هذه الجمل ...

كود

```
لفتح الاتصالcon.Open ()
لأغلاق الاتصالcon.Close ()
```

ولمعرفة حالة الاتصال ان كنا متصلين بالقاعدة أم لا سنستخدم الحالة State التابعة لكائن الاتصال والتي سينتج عنها عند الاستفسار ما يلي أن كان الاتصال في الاوضاع التالية :

الاتصال مفتوح[1] Open

الاتصال مغلق[2]Closed

جاري فتح الاتصال[3]Connecting

يتم تنفيذ امر استعلام على الاتصال[4]Executing

جاري الحصول على بيانات من سجلات مصدر البيانات[5] Fetching

وستأتي شفرة التأكد من حالة الاتصال بهذه الطريقة

كود

```
If (Con.State And ConnectionState.Open) <> 0 Then
التعرف إلى حالة الاتصال'HnHn
```

```
MsgBox("تم فتح الاتصال بنجاح")
Else
  MsgBox("تم اغلاق الاتصال")
End If
```

ويمكنك تغيير الوضع **Open** إلى إي من الأوضاع السابقة

سنأتي الأن على ذكر كائن الأوامر **Command**

ولتذكير فقط بما سبق ذكره عن الكائن

اقتباس

-[6]مجموعة البيانات **Command** يسمح هذا الكائن لكائن **DataAdapter** بتطبيق الأوامر على قاعدة البيانات

بعد تكوين الاتصال مع قاعد ة البيانات، ستأتي هذه الخطوة وهي ارسال جمل الاستعلام وهنا بأمكانك حصر البيانات التي تريد استعراضها كما تشاء أن كانت لديك فكرة في جمل الاستعلام **SQL** وستكون هذه الشفرة مع شفرة الاتصال وسيكون تعريفا بهذه الطريقة

كود

```
Dim cmd As New OleDbCommand()
```

أما أذا كانت قواعد البيانات من نوع **SQL Server .NET Data Provider** فسيكون تعريف الاتصال من نوع **SqlConnection** و بهذا الشكل:

كود

```
Dimcmd As As New SqlCommand()
```

وستكون شفرة الربط مع الاتصال بهذه الطريقة

كود

```
 HnHn'شفرة الاتصال بالقاعدة بشكل تام
Dim Con As New OleDbConnection(ConnString)
```

```
Dim cmd As New OleDbCommand()
Con.Open()
cmd.Connection = Con
```

تابع لعملية الاتصال (فتح الجداول والتعامل معها)



سنحتاج هنا إلى عنصرين رئيسيين لتكملة عملية التعامل مع البيانات وهما ...

[1] - OleDbConnection

[2] - DataSet

وهذا هو تعريفهما مع التعاريف السابقة الموضوعه في حدث Public Class Form1 وستأتي بشكل هذا

كود

```
Dim Con As New OleDb.OleDbConnection() REM HnHn :
 "تعريف كائن الاتصال"
تعريف كائن أمرDim cmd As New OleDbCommand() REM HnHn : "
 لتمرير الاستعلام ثم الاتصال"
تعريف كائنDim Dp As OleDb.OleDbDataAdapter REM HnHn : "
```

```
 تدفق البيانات بمعنى المصدر أو المزود"
Dim rs As New DataSet() REM HnHn : " تعريف كائن التعامل مع البيانات
 في الذاكرة"
Dim ConnString As String REM HnHn : " تعريف متغير لتخزين مسار'
 القاعدة"
```

وهكذا نكون كونا المجموعة الأساسية للتعامل مع البيانات بشكل تام وسنبدأ الآن بالاتصال بأحدى جداول القاعدة

[ملاحظة]:

ان كائن الاوامر OleDbCommand لا يصل إلى مصدر البيانات بشكل مباشر وانما يعتمد على كائن الاتصال والذي بدوره يصل إلى مصدر بيانات كما في هذه الشفرة والمذكورة لديك في المثال المرفق سابقا .

كود

```
con.Open()
cmd.Connection = con
con.Close()
```

وللأستفادة من كائن الأمر علينا بناء جمل ة أستعلام مع مراعاة نوعها أن كانت جملة استعلامية تقليدية أو جملة تنفيذية وللمعلومية ....

أن الجمل الاستعلامية هي التي لا تؤثر على سجلات قاعدة البيانات وانما تقوم بقراءة محتوياتها ونستخدم لها أمر SELECT

أما الجمل التنفيذية هي تلك الجمل التي تحدث تغييرا في سجلات جداول القاعدة بشكل التالي UPDATE ،INSERT INTO ، أو DELETE.

وغالبا ما يستخدم مع الجمل التنفيذية أمر ExecuteNonQuery() وذلك كنوع من الحصر لعدد السجلات التي تأثرت بالعملية ومثال ذلك العملية التالية ..

كود

```
Dim SQL As String = "UPDATE Emp SET Slary = 5000
WHERE NoEmp = 10001 "
Dim cmd As New OleDbCommand(SQL, con)
cmd.ExecuteNonQuery()
```

الأن سنأتي على عملية الاتصال بجدول البيانات وسحبها منه بشكل التالي

- [1]سنقوم بتعريف متغيير لوضع جملة الاستعلام بداخله كما يلي

كود

```
Dim SQL As String = "SELECT * FROM HnHnEmp"
```

- [2]سنقوم بوضع الاستعلام مع الاتصال في محول البيانات **OleDbDataAdapter**

كود

```
Dp = New OleDb.OleDbDataAdapter(SQL, Con
```

- [3]سنقوم بنقل البيانات لتعامل معها من دون اتصال بوضعها في**DataSet**

كود

```
Dp.Fill(rs, "HnHnEmp")
```

- [4]أخيرا سنغلق القاعدة لتعامل معها من دون اتصال وستصبح الشفرة بشكل نهائيا بشكل التالي .....

كود

```
Con.ConnectionString = ConnString REM HnHn : " الاتصال
"بمسار القاعدة
 Con.Open() REM HnHn : فتح الاتصال
```

```
SQL = "SELECT * FROM HnHnEmp" REM HnHn : "وضع"
Dp = New OleDb.OleDbDataAdapter(SQL, Con) REM HnHn : "تدفق البيانات مع الاتصال في  المتحول"
Dp.Fill(rs, "HnHnEmp") REM HnHn : "التعامل مع سجلات الجدول في الدات سيت"
Con.Close()
```

لم يتبق الآن سوى أظهار البيانات في الحقول المخصصة لها وستأتي الطريقة بشكل التالي

كود

```
TextBox1.Text = rs.Tables("HnHnEmp").Rows(0).Item(0)
```

وتفصيلها كما يلي .....

TextBox1.Text : الحقل الذي سيرتبط مع الحقل في الجدول

rs.Tables("HnHnEmp"). : اسم الجدول المدرج منه اسم الحقل المرتبط بالتيكس بوكس

Rows(0). : تمثل الصف الخاص بالحقل المرتبط وعند تغيير الرقم الذي بداخله يتم الانتقال لسجل التالي والعكس

Item(0). : تمثل عنصر الوصول إلى الفئات المحضونة من الكائن الرئيسي DataSet وهي خاصية افتراضية للكائن DataRow لذا يمكنك تجاهلها ان اردت .

وهكذا مع بقية الحقول بالشكل التالي وفق المثال المرفق....

كود

```
REM HnHn : اظهار بيانات الجدولفي الحقول المخصصة
TextBox1.Text = rs.Tables("HnHnEmp").Rows(0).Item(0)
TextBox2.Text = rs.Tables("HnHnEmp").Rows(0).Item(1)
TextBox3.Text = rs.Tables("HnHnEmp").Rows(0).Item(2)
TextBox4.Text = rs.Tables("HnHnEmp").Rows(0).Item(3)
```

```
TextBox5.Text = rs.Tables("HnHnEmp").Rows(0).Item(4)
TextBox6.Text = rs.Tables("HnHnEmp").Rows(0).Item(5)
TextBox7.Text = rs.Tables("HnHnEmp").Rows(0).Item(6)
```

عملية التنقل بين السجلات

في بيئة النت اختلف الأمر واصبحت عملية التنقل بهذا الشكل للسجل التالي

كود

```
Me.BindingContext(rs.Tables("HnHnEmp")).Position += 1
```

وللسجل السابق

كود

```
Me.BindingContext(rs.Tables("HnHnEmp")).Position - =
1
```

والسجل الأول سيكون بهذا الشكل

كود

```
Me.BindingContext(rs.Tables("HnHnEmp")).Position = 0
```

والسجل الأخير سيكون بهذا الشكل

كود

```
Me.BindingContext(rs.Tables("HnHnEmp")).Position =
Me.BindingContext(rs.Tables("HnHnEmp")).Count - 1
```

وهذا بشكل عام ولكن في مثالنا سيتم عمل التنقل بين السجلات بالطريقة التالية

- [1]سنقوم بوضع متغيير على مستوى الفوم وسيكون من نوع رقم لتخزين قيمة او رقم السجل

بداخله كما يلي

كود

```
Dim Rec As Integer
```

- [2]سنقوم بعمل أجراء بأسم

ViewRecordوسيحوي السجلات المرتبطة بالحقول وستكون قيمة Rows(0) هي المتغيير

Recالذي قمنا بتعريفه للسجلات بشكل هذا

كود

```
Rows(Rec)
```

وسيكون الأجراء بشكل هذا

كود

```
Private Sub ViewRecord()
 اظهار بيانات  الجدولفي الحقول المخصصةREM HnHn :
TextBox1.Text =
rs.Tables("HnHnEmp").Rows(Rec).Item(0)
TextBox2.Text =
rs.Tables("HnHnEmp").Rows(Rec).Item(1)
TextBox3.Text =
rs.Tables("HnHnEmp").Rows(Rec).Item(2)
TextBox4.Text =
rs.Tables("HnHnEmp").Rows(Rec).Item(3)
TextBox5.Text =
rs.Tables("HnHnEmp").Rows(Rec).Item(4)
TextBox6.Text =
rs.Tables("HnHnEmp").Rows(Rec).Item(5)
TextBox7.Text =
rs.Tables("HnHnEmp").Rows(Rec).Item(6)
'HnHn : Total Record
```

```
Label9.Text = Rec & Space(2) & "OF" & Space(2) &
rs.Tables("HnHnEmp").Rows.Count
End Sub
```

- [3]سنقوم الآن بتكوين عملية التنقل للسجلات بشكل التالي

السجل التالي

كود

```
Private Sub ButtonNavNext_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ButtonNavNext.Click

REM HnHn : Next
Rec = Rec + 1
Call ViewRecord()
End Sub
```

السجل السابق

كود

```
Private Sub ButtonNavPrevious_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ButtonNavPrevious.Click

REM HnHn : Previous
Rec = Rec - 1
Call ViewRecord()
End Sub
```

السجل الأول

كود

```
Private Sub ButtonNavFirst_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ButtonNavFirst.Click

REM HnHn : First
Rec = 0
Call ViewRecord()
End Sub
```

السجل الأخير

كود

```
Private Sub ButtonNavLast_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ButtonNavLast.Click

REM HnHn : Last
Rec = rs.Tables("HnHnEmp").Rows.Count - 1
Call ViewRecord()

End Sub
```

ولتنقل بأكثر أمان سنضع بعض الشروط لتحقق من السجل أن كان في الأول أو في الأخير كما يلي :

الانتقال لسجل التالي

كود

```
Private Sub ButtonNavNext_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
```

```
ButtonNavNext.Click

REM HnHn : Next
Rec = Rec + 1
HnHn 'التحقق من المتغيير اذا كان اكبر من مجموع عدد السجلات الكلي واعادة تخزين اخر قيمة :
If Rec > rs.Tables("HnHnEmp").Rows.Count - 1 Then
MsgBox("لا توجد سجلات للأنتقال إليها")
Rec = rs.Tables("HnHnEmp").Rows.Count - 1
Exit Sub
Else
Call ViewRecord()
End If

End Sub
```

الانتقال لسجل السابق

كود

```
Private Sub ButtonNavPrevious_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonNavPrevious.Click

REM HnHn : Previous
Rec = Rec - 1
HnHn 'التحقق من المتغيير اذا كان اقل من الصفر واعادة تخزين اخر قيمة :
If Rec < 0 Then
MsgBox("لا توجد سجلات للأنتقال إليها")
Rec = 0
```

```
Exit Sub
Else
Call ViewRecord()
End If
End Sub
```

****

لخدمات تصميم وبرمجة المواقع

programmer4ever@yahoo.com

00201063879624

****



عملية الأضافة والتحديث

الأضافة

سأتحدث هنا بطريقتين لأتمام عملية الاضافة وهما

[1] -استعمال خاصية DataRow

[2]-استعمال جملة استعلام INSERT INTO

DataRow

حيث انها تمثل سجل كامل من سجلات الجدول ولن نستطيع التعامل معها بإنشاء كائن باستخدام New من الفئة DataRow بمعنى ان تعريف المتغير بهذه الطريقة الخاطئة لن يفيد في عملية التعامل مع الفئة

كود

```
Dim dRow As New DataRow
```

وأنما يتم ذلك بطريقتين:

- [1] بتعريف سطر جديد NewRaw()

- [2] تعريف مصفوفة من النوع Object

ولن اشرح سوى طريقتين كما ذكرت سابقا وهما الأكثر استخداما في اعتقادي وايضا لكي لا تتشعب الامور بتفاصيل اكثر حيث اني احاول ان اركز على الامور الاكثر فائدة وتدول وكلن له طريقته......

اذا سنشرح

الطريقة الأولى باستخدام NewRaw() التابعة للفئة DataRow

كود

```
Dim dRow As DataRow = NameTabel.NewRow()
```

ملاحظة NameTabel : هو الجدول (مصدر البيانات)

وهذه هي شفرة عملية أضافة سجل بـالفئة DataRow والخاصية NewRaw()

كود

```
REM : HnHn كائن للإضافة سطر جديد لمجموعة البيانات
Dim dRow As DataRow

REM : HnHn اضافة سطر جديد في الجدول لتخزين البيانات فيه
dRow = rs.Tables("HnHnEmp").NewRow()
```

```
REM : HnHn : عملية نقل البيانات من الحقول إلى حقول الجدول
' أما عن طريق تحديد اسماء الحقول التي في الجدول  إو عن طريق الفهرسة

REM : HnHn : dRow.Item("LastName") = TextBox2.Text
REM : HnHn : OR Index
dRow.Item(1) = TextBox2.Text
dRow.Item(2) = TextBox3.Text
dRow.Item(3) = TextBox4.Text
dRow.Item(4) = TextBox5.Text
dRow.Item(5) = TextBox6.Text
dRow.Item(6) = TextBox7.Text
```

الطريقةالثانية باستخدام

INSERT INTO

جمل ال SQL هي احدى طرق عملية الاضافة وستكون شفرة الاضافة مسنودة إلى متغير من نوع كوماند OleDbCommandوقد قلنا سابقا أن هذه الفئة تؤدي مهمة تطبيق الاوامر على القاعدة وها نحن نستخدمها الآن في تطبيق جملة SQL من آجل عملية الأضافة حيث سنحتاج منها العنصر CommandTextوستكون الشفرة على سبيل المثال كما يلي ..

كود

```
Rem HnHn : 'تعريف متغير واسناده إلى كوماند ليقوم  بعملية الحفظ لجملة الاضافة
Dim SavInto As New OleDb.OleDbCommand

REM HnHn : INSERT جملة الاضافة بـ
SavInto.CommandText = "INSERT INTO HnHnEmp (NameField)
values ('" & TextBox1 & "') "
```

حيث = NameField سنكتب بدلا عنه اسماء الحقول الموجودة في الجدول وتفصل بينهما الفاصلة (,)وسنكتب نظيرها من الحقول اللي على الفورم بنفس التسلسل TextBox1 , TextBox2 وهكذا ...مع مراعاة علامات التنصيص ' أذا كان الحقل في الجدول من نوع نص وبدون علامات تنصيص أذا كان نوع الحقل رقم وعلامة # اذا كان نوع الحقل تاريخ وبناء على المثال المرفق لديكم سنكتب الشفرة كما يلي ...

كود

```
 Rem HnHn  :'تعريف متغير واسناده إلى كوماند ليقوم  بعملية الحفظ لجملة الاضافة
Dim SavInto As New OleDb.OleDbCommand

 REM HnHn  :  INSERT جملة الاضافة بـ

SavInto.CommandText = "INSERT INTO HnHnEmp
(LastName,FirstName,BirthDate,Address,Mobail,Notes) values
('" & TextBox2.Text & "','" & TextBox3.Text & "',#" &
TextBox4.Text & "#,'" & TextBox5.Text & "'," &
TextBox6.Text & ",'" & TextBox7.Text & "') "
```

تحديث البيانات

وهكذا عرضنا ثلاثة طرق لعملية الاضافة والموضوع لم ينتهي بعد حيث لابد من بعد عملية الاضافة تأتي عملية تحديث البيانات وهذا ما سنشرحه الآن مع كل طريقة من الطرق السابقة كيف سيتم تحديث البيانات فيها وسنبدأ بـــ

تحديث بيانات الطريقة الأولى NewRaw() : الخاصة بالفئة DataRow

وستأتي عملية تحديث البيانات هنا بتعريف متغير من نوع الفئة **DbCommandBuilder** الخاص بالعمليات التي تحدث على السجلات من جمل الاستعلام **SQL** ثم يتم اسناد الكائن للمحول **DataAdapter** ومن ثم فتح الجدول وأضافة سجل فيه مدرج في حقوله البيانات المسنده له ومن ثم اسناد السجل بحقوله للمحول في عملية تحديث للبيانات وتأتي الشفرة النهائية في هذه المرحلة من عملية الاضافة بهذه الطريقة كما يلي...

كود

```
Dim CmdB As New OleDb.OleDbCommandBuilder(Dp)
'-------------------------
REM : HnHn كائن للإضافة سطر جديد لمجموعة البيانات
Dim dRow As DataRow

REM : HnHn اضافة سطر جديد في الجدول لتخزين البيانات فيه
dRow = rs.Tables("HnHnEmp").NewRow()

  REM : HnHn : عملية نقل البيانات من الحقول إلى حقول الجدول
  ' أما عن طريق تحديد اسماء الحقول التي في الجدول إو عن طريق الفهرسة

REM : HnHn : dRow.Item("LastName") = TextBox2.Text
REM : HnHn : OR Index

dRow.Item(1) = TextBox2.Text
dRow.Item(2) = TextBox3.Text
dRow.Item(3) = TextBox4.Text
dRow.Item(4) = TextBox5.Text
dRow.Item(5) = TextBox6.Text
dRow.Item(6) = TextBox7.Text

  REM HnHn : اضافة السجل للجدول
```

```
rs.Tables("HnHnEmp").Rows.Add(dRow)

 REM HnHn  :عملية التحديث في قاعدة البيانات
Dp.Update(rs, "HnHnEmp")
MsgBox("تمت عملية الاضافة والحفظ في قاعدة البيانات  بنجاح")
```

تحديث بيانات الطريقة الثانية INSERT INTO :

وفي هذه المرحلة تتم عملية تحديث البيانات بالخطوات التالية

- [1]تعريف المتغير من نوع أمر [OleDbCommand]

- [2]تحديد مسار الاتصال واسناده للمتغير

- [3]تحديد نوع الامر وسيكون من نوع نص .. لأننا نحن من سيقوم بكتابة الاستعلام يدويا

- [4]كتابة جملة الاستعلام لأضافة الحقول للجدول

- [5]فتح الاتصال لأتمام العملية النهائية

- [6]حصر السجلات التي ستتأثر بعملية الأضافة

- [7]اغلاق الاتصال

والشفرة ستكون كالتالي ....

كود

```
 REM HnHn  :تعريف متغير واسناده إلى كوماند ليقوم بعملية الحفظ لجملة الاضافة
Dim SavInto As New OleDb.OleDbCommand

 REM HnHn  :تحديد مسار الاتصال للمتغير
SavInto.Connection = Con

 REM HnHn  :تحديد نوع الأمر وسيكون من نوع نص لأننا سنقوم بكتابة الاستعلام يدوي
```

```
SavInto.CommandType = CommandType.Text

REM HnHn : جملة الاضافة بـ INSERT
SavInto.CommandText = "INSERT INTO
HnHnEmp(LastName,FirstName,BirthDate,Address,Mobail,Notes)
values ('" & TextBox2.Text & "','" & TextBox3.Text & "',#"
& TextBox4.Text & "#,'" & TextBox5.Text & "'," &
TextBox6.Text & ",'" & TextBox7.Text & "') "

REM HnHn : فتح الاتصال لعملية الاضافة
Con.Open()

REM HnHn : لحصر عدد السجلات التي تأثرت بعملية الاضافة
SavInto.ExecuteNonQuery()

REM HnHn : اغلاق الاتصال
Con.Close()

MsgBox("تمت عملية الاضافة والحفظ في قاعدة البيانات بنجاح")
```

****

لخدمات تصميم وبرمجة المواقع

programmer4ever@yahoo.com

00201063879624

****